

REMARKS

This AMENDMENT UNDER 37 CFR 1.111 is filed in reply to the outstanding Office Action of April 5, 2004, and is believed to be fully responsive thereto for reasons set forth below in greater detail.

These REMARKS are organized into the following six sections.

- 1) Summary
- 2) Major distinguishing features of the present invention over Ogawa, and the inventor's response to the Examiner's comments related to Ogawa as prior art.
- 3) Major distinguishing features of the present invention over Colley, and the inventor's response to the Examiner's comments related to Colley as prior art.
- 4) Major distinguishing features of the present invention over Suzuki, and the inventor's response to the Examiner's comments related to Suzuki as prior art.
- 5) Major distinguishing features of the present invention over the rest of the cited prior art.
- 6) Revisions to the claims.

1) Summary

The present invention is distinctly different from the prior art (Colley, Ogawa, Suzuki etc). Sections 2 to 5 of these REMARKS comment upon the differences between the inventor's method and each of the prior art.

Moreover, the claims have been revised to remove any traces that would seem to suggest that the idea of 'caching', the concept of 'linked lists', or the terminology of 'dirty blocks' by themselves are significant aspects of the present invention. The claims are now directed to the new caching algorithm of the present invention.

2) Ogawa et al.

With respect to the objections raised by the Examiner in para 8, 9, 10, 11, 12, 13 of the Office Action, it is first noted that Ogawa uses the term 'sector' to mean logical file-system sectors and not flash sectors (which in Ogawa's terminology are termed 'erase blocks'). In Ogawa's semantics, the erase blocks (or actual flash sectors) are assumed to contain multiple 'sectors' (please read lines 10 to 20, page 8 of Ogawa). Secondly, Ogawa uses a mapping of logical-to-flash sectors in his method - which is a totally different approach from that of the present invention. In fact, page 1 of the present patent application (lines 26 to 29) lists certain disadvantages of methods like the one described by Ogawa (example, if the file-system data is uploaded from an embedded device that uses flash memory and moved to a customer support site, standard file system utilities cannot be used to debug/repair the device).

With respect to the objections by the Examiner in para 12 and 13, where he states that the change flag in Fig 12 of Ogawa anticipates some of the claims: In the Ogawa algorithm (that too only for sectors that have to be identified by outside code as belonging to FAT), Ogawa uses a 'least recently used' (LRU) method, instead of a weightage dependent on dirtied blocks and sector switches, as is used by the present invention. Also, the semantics of a dirtied block as used by Ogawa is different from that used by the present invention in the following respects: a) His 'dirtied block' is a yes/no flag, while the present invention uses an integer whose value is calculated using the method described in page 7 (line 1 to 4) of the present patent application. b) His use of the 'change flag' field (called dirty block field by the examiner) is not used for the actual working of his algorithm, but as an optimization (Please see lines 21 to 31, page 14 of Ogawa, where he says - "If the change flag is false, the control is... Although it seems strange to write data during the read procedure, to improve the efficiency, it is preferable to write....").

The present invention uses the term 'dirty block count' as a weightage value to determine which flash sector cache node is a good candidate to be freed. In fact, this weightage value is dynamically re-calculated and it is possible that constituent blocks of the said flash sector are deemed non-dirty (see step 3 below) even if they contain data that differs from the corresponding flash memory locations.

To briefly describe the semantics of the dirty block field as used by the present invention:

1. This field is initially zero.
2. This field is incremented when there is a write access to a file-system block within the cached sector, as long as the said block has not caused the count to be incremented so far.
3. This field is reset to zero if the number of sector switches suffered by that sector crosses a threshold. At this point, all the blocks in the said sector are also deemed to be non-dirty, even though they might contain data that differs from the corresponding flash memory locations. This will give preferential treatment to this sector and will let it live longer in cache.
4. Steps 2 and 3 are repeated.

Since the dirty block count is actually a function of the number of dirtied file-system blocks in the sector as well as the number of sector switches suffered by it, to be more accurate, the name of that field has been changed to 'sector weightage' throughout the present patent application.

Also, Ogawa's Figure 12 (linked list) is very different from Figure 4 of the present application, which is actually a list of data structures recording certain characteristics of data access to that sector as well as containing a cached copy of the data present in that sector.

With respect to objections in para 14, 15 and 16 of the Office Action, where the Examiner says that Ogawa anticipates persistent sector storage - ("the cache is reserved for FAT, the cache persistently stores FAT information" etc): The present patent application explains a heuristic approach to automatically detect the location of file system meta data irrespective of the file system used, and not based upon the mere fact that metadata needs to be in persistent storage. In direct contrast, Ogawa's method assumes that it knows whether a particular sector contains metadata (FAT). (Please see lines 9 to 13 in Page 14 of Ogawa - "In step S1502, whether the N sector is a FAT is checked."). In other words, the emphasis of the claims herein is not in the fact that persistent storage treatment be given to the meta data sectors, but in the

development of a heuristic to automatically detect the meta-data sectors in a file-system-independent manner and offer preferential treatment to those sectors.

3) Colley et al.

The Examiner lists Colley et al. - Col. 63,43-53 and Col. 64,48-50 – in pages 2 and 3 of the Office Action.

Some form of the method described by Colley (63,43-53) is generally used by most UNIX-like Operating Systems at the file-system level. In fact, the present patent application anticipates (in fact, assumes) that the operating system would use the kind of algorithms described by Colley at the file-system buffer-cache layer. (see page 4 of the present patent application, lines 20-25, where we say 'the algorithm assumes that the embedded operating system uses file-system buffer caching algorithms, such as those generally used by UNIX systems'). This technique is meant to be used in tandem with a system implementing a scheme as that described by Colley. Schemes similar to the one described by Colley are suitable for implementation at a higher layer (file-system level) in the Operating System and are unsuitable if used alone, to manage memory having the characteristics of flash memory, while the one developed by the present invention is meant for implementation at a low level flash device driver layer.

The file-system buffer cache algorithms, like the one described by Colley, do not aim to minimize flash sector erases, rather it is aimed to minimize traffic at the granularity of files - the files can be fragmented across the storage medium and across flash sectors – and cannot be used as a technique for data management in flash memory.

To point out a difference, Colley (63, line 46) says 'the least recently used buffer is used for a new sector'. However, the present invention uses a technique of selecting the sector with a maximum weightage which is a function of the number of dirtied blocks and the number of sector switches. Note that these two approaches are almost directly opposite to each other. This is partly due to the fact that the algorithm assumes that the nature of the write data streams

emanating from the operating system will be tempered by buffer-cache algorithms (like the one described by Colley) - Please see page 6, line 27 of the present patent application. Also the semantics of the word 'sector' as used by Colley (hard disk sector with no implication that a write to any location in the sector can be done only if that location is previously erased, and the erase can be done only at the granularity of an entire sector) is different from the word 'sector' as used by the present invention.

4) Suzuki et al.

The Examiner lists Suzuki et al. on page 6 of the Office Action.

Suzuki's approach to data management in flash is fundamentally different from the approach of the present invention. The core idea in Suzuki's approach is a virtual-to-real mapping of flash sectors and usage of a sector with the least erase count (which aims actually to perform wear-leveling or an attempt to uniformly use all sectors of the flash), to decide on the mappings. In this respect, Suzuki's approach resembles the approach used by Ogawa, and as pointed before is commented on in page 1 of the present patent application (lines 26 to 29), certain disadvantages of this method (example, if the file-system data is uploaded from an embedded device that uses flash memory, and moved to a customer support site, standard file system utilities cannot be used to debug/repair the device). Also, the algorithm does not use a 'least erase count' strategy.

Where Suzuki uses a linked list of cached data structure in Figure 6, the semantics of the fields as well as the method of caching are different. For example, the data area of each node in the linked list (Fig 6) corresponds to a block (of 512 bytes) and not a flash sector (like in the algorithm of the present invention) - Please see line numbered 38 in page 3). Also, the dirty state is a boolean (yes/no) field like in Ogawa, while the invention dirty field is actually an integer that stores the number of file-system blocks that have been dirtied. Also in the invention algorithm, it is possible that a block is deemed non-dirty even when its data differs from the corresponding flash memory locations - the term dirty block count used by me is more of a weightage value dynamically calculated to determine which sector is the best candidate to be freed. While

Suzuki's list is ordered in a LRU (least recently used) form, the invention method uses the above-explained weightage value. A node in the linked list (sc_info) of the present invention has eight fields as described in Figure 5 of the present patent application. Also the linked list is not a simple list of cached sectors, but is a much more complicated data structure containing characteristics of data access to each cached sector with pointers to the actual cached data (please see figure 4).

5) Rest of the cited prior art

Bruce et al (US 2002/0097594) :-

This mainly pertains to erasing sectors in multiple flash banks by asserting hardware command cycles so that the erase cycles are performed in parallel, and is unrelated to the method of the present invention. The word 'cache' as used by Bruce et al, has the generally known meaning of 'memory that is used to speed up the performance of a computer system having lower access speeds'. While the concept of caching itself is a well known technique, the present invention is a new caching algorithm to manage flash memory in a file-system independent manner.

Haines et al. (US 2002/0091895 A1) :-

Howard et al. (US 2002/0073276 A1) :-

Otterness et al. :-

Bruce et al :- (6000006)

DeWitt et al. :-

All the above references disclose methods that include some form of caching algorithms for data management of devices with certain characteristics. The method of the present invention is a different caching algorithm, that is tuned to be suited to store data on memory with the characteristics of flash memory, in a file-system independent manner.

For example, Haines et al. mainly pertains to the management of buffer memory in a disk drive. This caching algorithm is suitable for disk drives with no requirement that a write to a location has to be preceded by a time consuming erase cycle to a superset of the data that is to be written - and is not suitable for managing flash memory. The method of the present invention is a caching algorithm that is suitable for flash and similar other non-volatile memory with the above erase restrictions.

6) Revisions to the claims.

The claims have been revised to remove any traces that would seem to suggest that 'caching', the concept of 'linked lists' or the terminology of 'dirty blocks' are by themselves significant aspects of the present invention, and are now directed to the new caching algorithm.

Claims 1 to 4 have been re-phrased into a single claim 1 (currently amended):

Claims 5 and 6 have been re-phrased, please note steps 1 to 4 listed at the end of section 2 of these Remarks as a response to the Examiner's comments related to Ogawa, where the semantics of the word 'dirty block count' is explained.

Claims 7 to 10 remain as is.

Claim 11 has been rephrased to reflect the above changes to claim 1.

The emphasis of claim 20 has been shifted from the generally known fact that persistent treatment ought be given to meta data sectors, to the heuristic that has been developed by the present invention that automatically detects and offers cache persistence to sectors holding critical meta-data.

Claims 24 and 25 have been rewritten to elaborate on the 'code means'.

This application is now believed to be in condition for allowance, and a Notice of Allowance is respectfully requested. If the Examiner believes a telephone conference might

expedite prosecution of this case, it is respectfully requested that he call applicant's attorney at (516) 742-4343.

Respectfully submitted,

A handwritten signature in cursive script, appearing to read "William C. Roch".

William C. Roch
Registration No. 24,972

SCULLY, SCOTT, MURPHY & PRESSER
400 Garden City Plaza
Garden City, New York 11530
(516) 742-4343

WCR:jf